NAME

vodirectory − NVO Directory search client

SYNOPSIS

vodirectory [−<*flags*>] [<*keywords*> | <*term*>] [...]

OPTIONS

The *vodirectory* task accepts the following options:

−h,--help
Print a help summary to the terminal and exit. No processing is done following this flag.

−v,--verbose
Verbose output. The output will be more verbose than normal but exactly what is printed depends on whether other flags are enabled to changed the basic task behavior.

−-vverbose
Very-verbose output. Even more output.

The following flags control the major behavior of the task, i.e. the type
of output to present.

−c, --count
Print a count of matching records. For each of the search terms, a simple count of matching resources will be printed. A breakdown by service type will be printed if the verbose flag is enabled.

−e, --exact
Match exactly the search term (resolve mode only). The *term* in this case will normally be an IVO identifier, this flag can be used to match the identifier exactly rather than as a potential substring. This option will return an empty string if no exact match is found.

−l, --list List the full resource record for each matching resource. For each matching resource, all (well, most) of the information available about a resource will be printed to the screen. Additional information may be available from the Directory web interface.

−m, --meta
List the metadata for the data services associated with the resource. For each resource found, the table metadata (i.e. the column UCD values) associated with the resource will be printed. Currently only those DAL services supported by VOClient may be queried.

−r, --resolve
Resolve the search term to a specified resource. In *resolve mode* the search terms are assumed to be either the resource *ShortName* or *Identifier* and the match will be done using only these two fields in the Directory resource record. The default output is simply the *ServiceURL* for all matching records (i.e. resources where the search string is part of the *ShortName* or *Identifier* fields) unless the *-f* flag is given to select other fields.

−s, --search
Do a full search of the keywords and/or search terms. This is the default mode and will output a list of matching resources containing at a minimum the service *Type*, *Title*, *ShortName* and *Subject*. Additional information is printed depending on the *verbose* level.

Shorthand Convenience Options:

−I, --Id Print only the *Identifier* field for the resolved resource.

−L, --long
Suppress the linebreaks imposed for output fields that would wrap around the normal 80-character output and allow long lines. This allows tasks to parse the output more predictably.

−R, --Resolve
Print the *ShortName*, *ServiceType* and *Identifier* fields for the resolved resource.

−S, --SName
Print only the *ShortName* field for the resolved resource.

−T, --Title
Print only the *Title* for the resolved resources.

Constraint Options:

The list of allowed constraint strings is generally specified in the *Resource Metadata for the Virtual Observatory* document available on the IVOA document repository. These flags provide a convenient method to limit a result list to resources that explicitly specify a value for the given constraint.

−b <bpass> or --bandpass=<bpass>
Constrain the search to the specified bandpass string. The text argument following the -f flag will be matched against the *SpectralCoverage* field of the resource record. Directory resources allow a list of values, however only a single argument may be used to constrain the search. Permitted values include: "Radio", "Millimeter", "Infrared" (IR), "Optical", "Ultraviolet" (UV), "X-Ray" (XRay), and "Gamma-Ray" (GR). The match is case insensitive, values shown in parentheses may be given and will be substituted automatically.

−C <content> or --clevel=<content>
Constrain the search to the specified ContentLevel string. The Directory may contain data for a variety of intended audiences; Most often this constraint will be used to limit the results to 'Research' grade data.

−t <type> or --type=<type>
Constrain the search to the specified ResourceType string. The Directory records may contain any user-specified string, however unless you know specifically how a specific resource is defined, this constraint should use the values 'catalog' (for Cone services), 'image' (for SIAP services), 'spectra' (for SSAP services), 'table' (for Vizier tables), or a literal string which appears in the resource record. Additional aliases will be added as new data services are supported.

−-new <time> or --new=<time>
Constrain the search to those resources that have been newly created during the specified <time>. By default, <time> is an integer value assumed to be a number of days, the last character may contain one of the qualifying characters to change the time period: 'h' for hours, 'd' for days, refers to the last siz months. If no search term is given, all results for that period will be returned, otherwise only those results that match both the search terms and the time constraint will be listed.

−-updated <time> or --updated=<time>
Constrain the search to those resource records that have been updated during the specified time period.

Output Control Options:

−a, --all Output all matching records (default). When used with the constraint flags above, this flag will allow those constraint strings to be matched as a substring, e.g. using "-t siap" will exactly match resources with type

−f <fields> or --fields=<fields>
Output the specified fields from the resource record (used in Resolve Mode only). The list of available fields is given below, an 'INDEF' string is printed for invalid field names or when no information is available.

−o, --or Logically OR the search terms. By default, all terms will be used when matching resource records.

−n <index> or --index=<index>
Output only the results for the matchng *index*. Results are 1-indexed, i.e. the first result is index 1 (one).

DESCRIPTION

The *vodirectory* task provides a command-line interface to the *NVO Directory* at STScI/JHU. The task also provides a basic search capability for the Directory, as well as a "Resolve Mode" that can be used to lookup resource records given some familiar name (e.g. 'USNO-B1'). Constraint parameters allow the search to be restricted to resources declaring a specific type (the *-t* flag), spectral coverage (the *-b* flag), or content level (the *-C* flag).

Search terms may be provided on the command-line, in a filename specified on the commandline, or read from the standard input (e.g. redirected from a file or other command). Advanced users can submit a quoted ADQL string to access specific fields of a resource record. This is similar to using the "Advanced Search" capability on the NVO Directory web page. ADQL strings are required to be in double quotes when given on the command line, the quotes are needed when query strings come from a file. Standard SQL operators apply for comparison and boolean operations, the 'like' operator is used to match strings (which must be in single quotes), one or more '%' metacharacters may be used in the string to indicate a wildcard match.

In the default search mode, keywords given on the command line will all be used to match resource records. The *-o* flag may be used to logically OR the keyword terms, e.g. to allow a search of 'galaxy' or 'galaxies'. The minimal output provides the resource title, type, subject and the 'ShortName' that can be used in the resolve mode or be passed to other tasks such as vodata(1). Additional output can be had with the *-v* or *-vv* verbose flags. A simple count of the resources will be printed if the *-c* flag is set (e.g. the command "vodirectory -oc chandra spitzer" will print a count of how many records match each term rather than display them directly, without the -o flag a count of resources mentioning both keywords will be printed).

The "Resolve mode" is activated by the *-r* flag; In this mode the keywords will only be matched against the Directory *ShortName* and *Identifier* fields. The default output is simply the *ServiceURL*, adding the verbose flags will instead print the ShortName, ResourceType and Title (with "-v") or Description (with "-vv"). The user can select specific fields to be printed using the *-f* flag followed by a comma-delimited list of fields. The allowed fields are shown with the *-h* help flag.

The *-list* flag implies Resolve Mode and will cause all fields of the matching resource to be printed. Unless the *-a* flag is set, the search term will be matched exactly, otherwise it will be considered to be a substring of the ShortName or Identifier fields. For example, searching with the term '2mass' will list only 2MASS image service, but using the *-a* flag will list all services where '2mass' appears in the ShortName.

The *-meta* flag likewise assumes the command line arg is a resource ShortName to be resolved and will query the DAL service associated with it using a *FORMAT=METADATA* query. The default position will be (0.0,0.0) with a search size of 0.1 degrees, the response will be a list of the column UCDs returned by the query (note that adding *-v* flags will likewise increase the VERBOSE of the query and may return additional columns).

RETURN STATUS

The task will exit with a status of 0 if at least one search term could be successfully resolved, otherwise the status will be 1.

VOCLIENT DAEMON PROCESSING

All VO-CLI tasks are built upon the VOClient interface an rely on a separate *voclientd* process to provide the VO functionality. The voclientd task is distributed as part of VO-CLI and will be started automatically by each task if it is not already running. If problems are encountered, you may want to manually start the voclientd in a separate window before running the task so you can monitor the output for error messages.

RESOURCE CACHING

Directory resolution is a common activity of VO-CLI tasks and so results will be cached in the $HOME/.voclient/cache/regResolver directory based on the search term, service type and bandpass parameters. Defining the *VOC_NO_CACHE* environment variable will cause the task to ignore the cache.

EXAMPLES

1) Get a count of all the SIAP services available in the Directory, then list more information about each one:

```
% vodirectory -count -t image
142
% vodirectory -rv -t image
```

2) Find all catalog (i.e. Cone) services using the search words 'radio' and 'galaxies':

```
% vodirectory -t catalog radio galaxies
```

3) Print the full resource record of the GSC2.2 catalog at STScI:

```
% vodirectory -list GSC2.2
```

4) Find all services with radio data of Abell clusters. Then print the full description of the first record associated with one of the matching Vizier tables:

```
% vodirectory -b radio abell
% vodirectory -rvv -n 1 J/A+A/446/97/tab
```

5) Find all image services that have WFPC data:

```
% vodirectory -v -t image wfpc
```

6) Print a breakdown of VO services having Keck data:

```
% vodirectory -cv keck
keck        122   (Cat: 2  Tab: 117 SNode: 1  Other: 2)
```

7) Print a count of services having all of Chandra, HST and Spitzer data, then break it down by each mission:

```
% vodirectory -c chandra hst spitzer
chandra hst spitzer       3
% vodirectory -co chandra hst spitzer
chandra              323
hst                  443
spitzer               31
```

8) Print the column metadata returned by the GSC2.2 service:

```
% vodirectory -meta gsc2.2
```

9) Use the ADQL query format to find services in which HST was the used, and not simply a match of 'HST' in the resource record:

```
% vodirectory "Facility like 'HST'"
```

Note that use assumed knowledge of the Directory being queried, specifically that there exists a 'Facility' field with this information and that the syntax of the query requires the string to be in quotes.

10) Use the ADQL query format to find services in which 'Keck' appears in the Title of the resource:

```
% vodirectory "Title like '%Keck%'"

        or

% cat query.txt
Title like '%Keck%'
% cat query.txt | vodirectory
```

Note that here we use the '%' operator around the string so that we perform a substring match on the entire title. As before, the ADQL string itself must be enclosed in double quotes.

11) Find all resources that are newly registered in the last 3 months, then find only those resources dealing with "cool stars", and finally just print a count of resources updated in the last year:

```
% vodirectory --new 3m
% vodirectory --new 3m cool stars
% vodirectory --updated 12m --count
```

BUGS

Some services don't repond properly to the metadata query and will print a "no attributes found" message.

Revision History

June 2007 - This task is new.

Author

Michael Fitzpatrick (fitz@noao.edu), July 2007

SEE ALSO

voclient(1), voclientd(1), vosesame(1), vodata(1)